Title of the Prior Art

Japanese Published Patent Application No. Sho.59-58580

Date of Publication : April 4, 1984

Inventor : Takayuki Nakagawa et al.

Concise Statement of Relevancy

This reference discloses a masking vector arithmetic processor, in which generation of a vector mask and a conditional vector arithmetic processing are simultaneously carried out in parallel with each other. However, the vector arithmetic processor described in this reference comprises plural registers specifically for vector masks, and generates the vector mask by a circuit of a system which is different from a system for obtaining arithmetic processing target data. This vector arithmetic processor does not decide whether arithmetic is to be executed or not, in parallel with the process for obtaining the arithmetic processing target data, as the present invention defined in Claims 1 and 3. Further, this vector arithmetic processor does not decide the state of supplied source data in the pipeline, as the present invention defined in Claims 2 and 4.

According to Claims 1 and 3 of the present invention, when the arithmetic processing target data is obtained in an arithmetic processing step of executing an arithmetic processing, it is computed and decided whether the arithmetic is to be executed or not, in parallel with the process of obtaining the target data. Thereby, a pipeline processing of performing a conditional arithmetic of whether an arithmetic using input data is to be executed or not, without using a specific mask register, and retaining the result can be executed without interrupting the flow.

According to Claims 2 and 4 of the present invention, the state of supplied source data is decided in the pipeline, and the conditional arithmetic is performed depending on the state of the source data as the condition. Thereby, a pipeline processing of performing a conditional arithmetic of whether an arithmetic using the source data is executed or not, without using a specific mask register, and retaining the result can be executed without interrupting the flow. In addition, the overhead in which the condition for performing the conditional vector arithmetic is computed can be dispensed with.

⑲ 日本国特許庁 (JP)　　⑪ 特許出願公開

⑫ 公開特許公報 (A)　　昭59—58580

⑤1Int. Cl.³　識別記号　庁内整理番号
G 06 F 15/347　7056—5B

⑬公開 昭和59年(1984)4月4日

発明の数 1
審査請求 未請求

(全 8 頁)

⑤4マスク付きベクトル演算処理装置

⑳特　願 昭57—168358
㉒出　願 昭57(1982)9月29日
⑫発 明 者 中川貴之
国分寺市東恋ケ窪1丁目280番地株式会社日立製作所中央研究所内
⑫発 明 者 長島重夫
国分寺市東恋ケ窪1丁目280番地株式会社日立製作所中央研究所内
⑫発 明 者 阿部仁
秦野市堀山下1番地株式会社日立製作所神奈川工場内
⑪出 願 人 株式会社日立製作所
東京都千代田区丸の内1丁目5番1号
⑭代 理 人 弁理士 薄田利幸

明　細　書

発明の名称　マスク付きベクトル演算処理装置

特許請求の範囲

1. 2つ以上のベクトルマスク専用レジスタと、少なくとも1つのベクトルマスク専用演算器と、該ベクトルマスク専用レジスタへの書込み処理に並行して、同一のベクトルマスク専用レジスタからの読出し処理を行うための同期の手段と、ベクトルマスク専用レジスタ間での演算命令の処理にあたつて、要求されるマスク情報間での読み出し及び演算の同期手段と、マスク付きベクトル演算命令の処理にあたつて、要求されるベクトルデータ及びマスク情報の間での読出し及び演算処理の同期手段とを有する、マスク付きベクトル演算処理装置。
2. 該ベクトルマスク専用レジスタのうち、同一のものに対する書込み処理と読出し処理の間での該同期手段は、書込み命令に後続する、読出し命令の処理における、ベクトルエレメント毎の同期をとる手段を有する第1項の装置。
3. 該マスク付きベクトル演算命令において、該ベクトルデータの保持及び読出しに、ベクトルレジスタを有する第1項の装置。
4. 該マスク付きベクトル演算命令において、ベクトルレジスタ中のデータに対し、マスク付きの主記憶格納内の手段を有する第3項の装置。
5. 該マスク付きベクトル演算命令以外の演算命令の解読結果として選択する演算器として、マスク付き演算命令に用いる演算器を流用する手段を、ベクトルデータとベクトルマスクデータとの該同期手段と、ベクトルマスクデータの読出し手段の一部として有する第1項の装置。

発明の詳細な説明

〔発明の利用分野〕

本発明はベクトルデータを高速に演算する装置に関する。

〔従来技術〕

ベクトルプロセサのより高速な処理を達成するには、より多くの種類の処理をベクトルプロセサにより高速に処理可能とすることが課題となつて

いる。中でも、FORTRANプログラムで、ＩＦ文を含むＤＯループを高速に処理するには、より高度の処理装置が必要とされるが、第１図に示すようなＤＯループの従来技術による処理手順を第２図と第３図を用いて説明する。

第１図に示すＤＯループでは、式(1)にあるような論理演算の結果の値が"真"であり、かつ、式(2)に示す論理演算結果が"真"である場合に限り対応するインデクス値：Ｉを持つエレメント間で式(3)の加算及び代入を行い、式(1)か、式(2)のいづれかが成立しないようなインデクス値を持つエレメント間での式(3)の加算及び代入は行わない。このような処理をＮ組のエレメントについて繰り返す。

第２図は、エレメント数：Ｎが６で、オペランドＡ（１～６），Ｂ（１～６），Ｃ（１～６），Ｄ（１～６），Ｆ（１～６），Ｇ（１～６）に適当な数値を仮定し、第１図の演算を行なつた場合のデータの流れを示したものである。

第２図で値"Ｘ"はこれらの処理により変更を受けず、また、この処理に関与しないことを示す。

以下に、その処理手順を示す。

ステップ１：　オペランドＡ（１～６）とＢ（１～６）の対応するエレメント同志を、それぞれ比較してベクトルマスク：ＶＭ（１～６）を作成する。この場合、対応する両エレメントの値が一致して論理演算結果が"真"となるとき、ＶＭには値"１"を書き、一致しないときＶＭには値"０"を書くものとする。従つて本例ではＶＭ（１～６）の値はそれぞれ、０，１，０，１，１，０となる。

ステップ２：　次のステップ３で、ベクトルマスク：ＶＭを格納している、ベクトルマスクレジスタを、再度書込みに使用する為に、ＶＭ（１～６）の内容を、別のレジスタＳＲ１に退避させる。なお、このＳＲ１は、ベクトルマスク専用のレジスタではなく、保持データの全エレメント分を一括して処理することしか出来ない。その為に、このステップは、ステップ１が全て終了してから行われる。

ステップ３：　ステップ(1)と同様に第１図の式(2)の演算を行い、結果をＶＭ（１～６）に書込む。ここでも、論理演算値"真"に対して値"１"，"偽"に対して値"０"を対応させると、この場合のＶＭ（１～６）の値は、それぞれ０，１，１，１，０，１となる。

ステップ４：　次のステップ５で、全エレメント分の一括処理しか出来ない汎用の演算器を用いて汎用のレジスタ間の論理演算を行うため、ＶＭ（１～６）の値を汎用のレジスタＳＲ２に退避する。このステップもステップ２と同じ理由で、ステップ３が全て終了してから行われる。

ステップ５：　ＳＲ１とＳＲ２のビット毎の論理積をＳＲ３に求める。このステップはステップ４が終了してから行われる。

ステップ６：　ステップ５で求めたＳＲ３の値をＶＭ（１～６）に転送する。

ステップ７：　ステップ６で得られたＶＭ（１～６）の値にをもとに、第１図の式(3)に示す加算と代入を行う。その際、対応するベクトルマスクＶＭ（Ｉ）の値が"０"のエレメントについては演算の結果を無効とする。すなわち、その時のＥ（Ｉ）の値を変更しない。本例では、第２及び第４エレメントについての み演算結果が主記憶上のＥ（Ｉ）の値を変更するように動作する。

以上の処理の様子を、第３図のタイムチヤートで示す。タイムチヤートの縦軸は各ステップを示し、横軸は時間を示す。各番号は、そのエレメント番号の処理の最初のサイクルを示しており、ステップ間の時間的ずれは、起動時間のずれを示している。

以上に述べてきたように、従来技術による複雑な条件文を含むＤＯループのベクトルプロセサでの処理は、ベクトルマスクレジスタが１つしかない為、一括処理用のレジスタに移動して、一括処理による演算処理を行い、出来あがつた最終的なベクトルマスクを、再びベクトルマスクレジスタに転送しなければ、条件付きの演算処理が出来ない。従つて、第３図に見るように、従来技術ではベクトル処理が、ステップ１と、ステップ３、ス

テップ7に、それぞれ処理時間帯を分けられ、これらのステップ間では、処理の並列・高速化ができないという問題があつた。

一般に、ベクトルプロセサでは複数ベクトル演算の異なるエレメントについての処理を、同時並行に処理することで高速性を達成しているが、ステップ2，4，5，6の処理がベクトルプロセサ向きでない為に、第4図のように、ステップ1，3，7を並列に処理することを妨げている。

〔発明の目的〕

本発明の目的は、ベクトルマスクの生成、ベクトルマスク間の演算処理、及び条件付きベクトル演算処理を、全て同時並行して処理可能とするベクトル演算処理装置を提供することにある。

〔発明の概要〕

このため、本発明による装置では、

(1) 同時に書込み・読出し可能な複数のベクトルマスク専用レジスタ。

(2) ベクトルマスク専用レジスタ中のデータを入力として新しいベクトルマスクの値を算出する、1個以上のベクトルマスク専用演算器。

(3) ベクトルマスクの書込みと、ベクトルマスク専用演算器への読出しを同期させる手段。

(4) ベクトルマスクの書込みと、マスク付きベクトル演算処理用演算器への読出しを同期させる手段。

とを、新たに設けることにより、ベクトルマスクの生成，ベクトルマスク間演算，条件付きベクトル演算処理を、全て同時並行して処理することを可能としたものである。

ただし、本明細書では"演算処理"という言葉により、主記憶参照，ベクトル間の加減算等の演算，結果の主記憶への格納の全てのものをさし、本明細書の実施例ではマスク付き主記憶格納を例にとる。

〔発明の実施例〕

以下、本発明を実施例を参照して詳細に説明する。第5図は本発明の一実施例を示す。

本発明に直接関連しない装置部分は、通常のベクトルプロセサと同じ構成を有するものとし、その部分の説明は簡単にするに止める。

第5図において、101は主記憶装置、102は主記憶制御装置、103はスカラ処理装置であり、104はその一部であるスカラ命令制御装置、105はベクトル命令制御装置である。

スカラ命令制御装置104では主記憶制御装置102と信号線201を介して主記憶装置101から順次読出した命令を解読し、これが通常のスカラ命令であるときは、スカラ処理装置103にて通常のスカラ演算処理を行い、信号線202を介し結果を主記憶装置101に書込む。スカラ命令制御装置104で解読した命令が、ベクトル命令列の起動を指示する命令であれば、信号線203を介して、ベクトル命令列の主記憶装置101上での先頭アドレスと、処理すべきベクトルエレメント数と、起動信号をベクトル命令制御装置105へ渡す。引続き、ベクトル命令制御装置105は与えられたアドレスに従い、信号線201，204を介してベクトル命令列を主記憶装置101から順次読出し、ベクトル命令を解読し、命令中で指定された、レジスタ，演算器，メモリリクエスト等が使用可能な状態にあると判断した命令から、それぞれ信号線205，206，207を介して命令に必要なリソースを起動すると共に、同時に、処理すべきベクトルエレメント数を含めた制御情報を転送する。ベクトル命令のエレメント個々に対する処理は、ベクトルレジスタ制御装置110や、ベクトルマスクレジスタ制御装置120の送出する。1エレメント毎の処理の許可信号に従つて進められる。以下、第1図に示したDOループの処理を行う場合を例にとり、本発明によるベクトル命令の実行を説明する。本例では、ベクトルレジスタと呼ぶバッファ記憶を用いており、それぞれNエレメントからなる一連のデータ、A(1～N)，B(1～N)，C(1～N)，D(1～N)，E(1～N)，G(1～N)を信号線211～216をを介し、ベクトルレジスタ111～116にそれぞれ格納することが必要であるが、この間の処理は通常のベクトルプロセッサによる処理に従うものとする。

第１図の処理を実行するには、上記の処理を行う６つの命令の他に、Ａ（Ｉ）とＢ（Ｉ）を要素ごとに比較し、比較結果をベクトルマスクレジスタに書込む第７の命令。同じく、Ｃ（Ｉ）とＤ（Ｉ）の値の要素ごとの比較結果を第７の命令で書込んだのとは別のマスクレジスタに書込む第８の命令。第７の命令と第８の命令により得られる２つのベクトルマスク間で論理演算を行い、結果を第３のベクトルマスクレジスタに書込む第９の命令。Ｆ（Ｉ）とＧ（Ｉ）を要素毎に加算し、結果をベクトルレジスタに格納する第１０の命令と、第１０の命令によつて得られた加算結果を、第９の命令によつて得られたベクトルマスクが"１"をとる場合のみ（もしくは、"０"をとる場合のみ）、Ｅ（Ｉ）に相当するメモリアドレスに書込む、第１１の命令が用いられる。

以下第５図を用いて、第１図の処理を説明する。第７の命令に応答してベクトル命令制御装置は、演算器１３１に、ベクトルレジスタ１１１，１１２中の値を送り、式(1)の比較を行なう。この場合、等しいという関係が成立する場合には、値"１"を、成立しない場合には値"０"を、信号線２３１を介し、ベクトルマスクレジスタ１２２に書込むものとする。この比較演算は、１３１にパイプライン演算器を用いることで、１サイクルに１エレメントのピッチで進めることができる。第８の命令は、同様に式(2)の比較を行い、成立に対し"１"を、不成立に対しては"０"を、パイプライン演算器１３２から信号線２３２を介して、ベクトルマスクレジスタ１２３に書込む。

１２２と１２３に書込まれた値の間で、今度は第９の命令によつて論理積をとる演算が行われる。これは、従来技術で述べたような、一括処理しか出来ないレジスタ及び演算器ではなく、ベクトルマスクの専用レジスタ１２１～１２３、及び専用のパイプライン演算器１４１を用いているため、必要なデータが揃つたエレメント間では、ただちに論理積をとる処理に移るように同期される。

また、第９の命令で作成されたベクトルマスクの値を、第１１の命令で参照する場合も、メモリに格納するデータと、ベクトルマスクが揃つたエレメント間では、直ちに処理が行われるように同期される。

前者のベクトルマスクレジスタ間の同期機構を第６図で説明し、後者のベクトルマスクレジスタとベクトルレジスタの間での同期機構を、条件付きベクトル演算を例に、第７図で説明する。

第７，第８の命令で書込まれたベクトルマスクレジスタの値は、書込みが全エレメントにわたつて終了していれば、どこから読んでも良いが、途中のエレメントを書き込んでいる途中なら、書込む前の値を読むと、第１図の処理が正しく実行されない。この書込み済の範囲を知る目的で、例えば、比較演算結果を書き込んでから、論理積をとるため読出されるベクトルマスクレジスタ１２２（以下ＶＭＲ２と呼ぶ）に対応して、書き込み済みエレメント数が、どれだけ読み出しエレメント数を上回つているかを表える、アップダウンカウンタ３２２と、書込み中であるとか読出し中であるという状態を表えるレジスタ３２１を用意する。ＶＭＲ２から１エレメント読み出す為の許可信号４０４は、ＶＭＲ２が書込み中でない場合か、書込み中だがこれから読み出そうとするエレメントが書込み済である場合に発行される。カウンタ３２２は書込みエレメント数から読出しエレメント数（もしくは読出し予定エレメント数）を引いた差を保持しているので、このカウンタの値が正であることを示す信号４０１、ＶＭＲ２が書込み中かつ、読出し中であることを示す信号４０２、ＶＭＲ２が書込み中ではない、単純な読み出し状態にあることを示す信号４０３に、ＡＮＤ回路３２３、ＯＲ回路３２４の論理をとることで、信号４０４を得ることができる。

また、ＶＭＲ２とベクトルマスクレジスタ123（以下、ＶＭＲ３と呼ぶ）の間で演算をして良いのは、ＶＭＲ２も、ＶＭＲ３も、どちらも１エレメント読み出す許可信号が揃つている場合に限る。１エレメントの演算許可信号４０６は、ＶＭＲ２の１エレメント読出し許可信号４０４と、ＶＭＲ３の１エレメント読出し許可信号４０５をＡＮＤ

回路３０１を用いてＡＮＤすることにより得られる。この１エレメントの演算許可信号４０６によつてＶＭＲ２と３の読出しアドレス３０２の値が１追加され、未処理エレメント数を蓄えたカウンタ３０３の値が１でないなら、１減じて、次のエレメント処理へと進む。この論理積演算はパイプライン演算器１４１の各ステージ、１４１１～１４１３を経て、ベクトルマスクレジスタ１２１（以下ＶＭＲ１と呼ぶ）に書込まれる。この際、書込み側のアドレス３０４やアップダウンカウンタ３１２への加算は、３０６～３０７に示す同期フリップフロップを適当に追加することで、データの処理の同期がとられる。ＶＭＲ１に書き込んだエレメントデータを、必要な場合に、書込みと同時に読み出すには、アップダウンカウンタ312への＋１動作が行われてから、実際にデータが１２２へ書込まれるまでの時間差を、アップダウンカウンタ３１２の－１動作が行われてから、読出しが行われるまでの時間差に等しく設計することが必要である。本例ではそのために、信号407は３０７のフリップフロップ群をバイパスしてアップダウンカウンタ３１２へ送られている。この３０７のフリップフロップのうち最初のもの３０７１から、ＶＭＲ１までの信号の伝達時間と、３１２から、第６図では省略されているＶＲ１の読出しアドレス（第７図５０３）までの伝達時間を等しくすることによつて、ＶＭＲ１に書込まれたデータを直ちに読出すことが可能となつている。

第６図で、ＷＴと書かれているのは、それぞれのベクトルマスクレジスタへの書込み命令により発行される信号である。本例では、第１図の式(1)の処理の始まりと共に、信号４２１により、ＶＭＲ２の状態３２１を、書込み状態にセットし、アップダウンカウンタ３２２を０にリセットする。そして、１エレメント毎の書込み信号４２２により、書き込み数をカウントアップさせると共に信号４２３により、結果をＶＭＲ１に書込む。そして、全エレメント分の処理の終了をカウンタ303の値が１であるという信号４２４により検出し、書込み状態にあるという記録を３１１から、読出し状態にあるという記録を３２１～３３１から解除する。

以上の動作は、第６図で説明しているベクトルマスクレジスタ間演算によるＶＭＲ１の書き込み制御方式と同じため、第６図ではＶＭＲ２及びＶＭＲ３の書込み側の制御回路は省略してある。

第１図の式(1)，式(2)が全エレメント処理されるのを待たず、式(1)と式(2)のベクトルマスク間演算命令が起動されると、ＩＮＩＴ と書かれた信号２０７により、まず、ＶＭＲ１の状態３１１が書き込み状態にセットされると共に、アップダウンカウンタ３１２が０にリセットされるのは、先に述べた式(1)の処理と同様であるが、さらにＶＭＲ２及びＶＭＲ３の状態が、読み出し状態にセットされる。従つて本例で第１図のプログラムの処理を行つた場合、ＶＭＲ２は読み出しかつ書込み中の状態となり、信号４０１がオンで信号４０３がオフのため、書き込み済のエレメントのみについて読み出しの許可を与える制御を行う。

前述の式(1)の命令により１エレメントの書き込みがある度に、カウンタ３２２は＋１されるから、その結果を使うベクトルマスク間演算では、カウンタ３２２の値が正である限り、読み出し許可信号４０４を発行し、カウンタ３２２から－１する。例では、簡単の為に４０８により－１しているが、４０６により－１してもよい。

この４０８により－１した場合は４０６により－１した場合に比べ性能の劣化する場合があるが、この劣化は３９１～３９４にある回路により回避できる。信号４０６により－１する場合３９１～３９４の回路は不要である。以下では、簡単のため、ベクトルレジスタ，ベクトルマスクレジスタの読出し許可信号作成及び、演算許可信号作成時の３９１～３９４に相当する回路は図面中では省略する。

ＶＭＲ２と同様にＶＭＲ３も、読み出し許可信号を作成し、両者のＡＮＤ条件をとつた４０６が最終的な演算許可信号となつて読み出しアドレス３０２の値を進め、読み出し用セレクタ３２６，３３６を介して、ベクトルマスク専用演算器305

にデータを送ると共に、書込み許可信号としてカウンタ３１２の値に＋１を行い、書込みアドレス３０４の値を進める。ここで、演算器１４１をベクトルマスクレジスタ間演算専用とすることで、僅かなコストの追加により、処理の高速化が可能となつている。第５図では、ベクトルレジスタ用演算器３個、ベクトルマスク専用演算器１個の構成となつているが、ベクトルマスク専用演算器は入力データ巾が、前者の３２～６４ビット／エレメントに対し、後者は１ビット／エレメントであり、コストもそれに順じて少なく出来る。

第６図の制御により、第７～８の命令と第９の命令が同期されることが判るが、第７図は、第11の命令と第１２の命令の同期方法を示したものである。ＶＲ２とＶＲ３の読み出し制御回路は、第６図のベクトルマスクレジスタのＶＭＲ２とＶＭＲ３のそれと同様のものを採用している為、ここでは説明を省く。ＶＭＲ１の読み出し制御回路と第６図の読出し制御回路との相違は、５０２のフリップフロップを条件付き命令でないことを記憶させるより、新たに設け、その場合、演算器等に送るベクトルマスク値６０３を、信号６０２を介して常に"１"にしていることである。このフリップフロップにより、条件付きベクトル演算命令以外にも、同じ演算器５０５を使用できる。

また、第６図でＶＭＲ２とＶＭＲ３の読出し許可信号間のＡＮＤをとつた信号４０６に相当して、第７図での演算処理許可（本例では１エレメントの主記憶格納許可）信号６０６は、ベクトルマスクレジスタＶＭＲ１の読出し許可信号６４４と、ベクトルレジスタＶＲ１の読出し許可信号６３４と、主記憶書込み制御回路１０２１からの受付け許可信号６０１との間でＡＮＤをとつたものとしている。条件付き命令以外の主記憶書込み命令の処理の場合は、信号２０７によりレジスタ５０２に１を格納し、信号６０２とＯＲ回路５４３により、信号６４４を常に１に設定する。

〔発明の効果〕

以上のようにして、第４図のタイムチャートに示したように従来は３Ｎ＋３サイクルかかつた処理を、本特許によりＮ＋３サイクルで処理できる。この短縮された２Ｎサイクルの内訳は、ベクトルマスク生成の比較演算命令と、ベクトルマスク間の論理積演算命令との間での処理の非並列化による従来方式での損失がＮサイクル、論理積演算命令と、条件付きベクトル演算処理（本例では条件付き主記憶格納）命令との間での処理の非並列化による従来方式での損失がＮサイクルであり、両損失の改善には、第６図と第７図に示した２通りの同期機構がそれぞれ寄与している。両改善とも、専用のベクトルマスクレジスタと演算器を使用せずに、通常のベクトルレジスタを拡張して使用しても可能であるが、専用化することにより、同じ性能を得る上で、コスト的に３２～６４倍有利であることが明らかである。

図面の簡単な説明

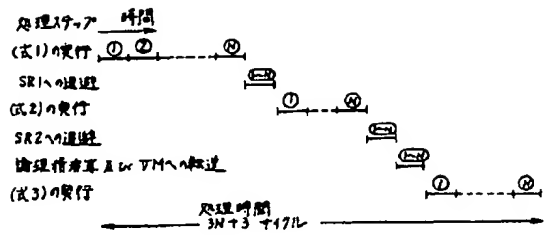
第１図はマスク付きベクトル処理を含むFORTRANプログラム例。第２図は従来技術による第１図プログラムの処理手順。第３図は、従来技術による図１の処理のタイムチャート。第４図は本発明による図１の処理のタイムチャート。第５図は、本発明による処理装置の一構成例を示す。第６図は、ベクトルマスクレジスタ間の演算実行制御の一実施例、第７図は条件付きベクトル主記憶格納処理実行制御の一実施例を示す。１２１～１２３はベクトルマスク専用レジスタ。１４１はベクトルマスク専用演算器。

代理人　弁理士　薄田利幸

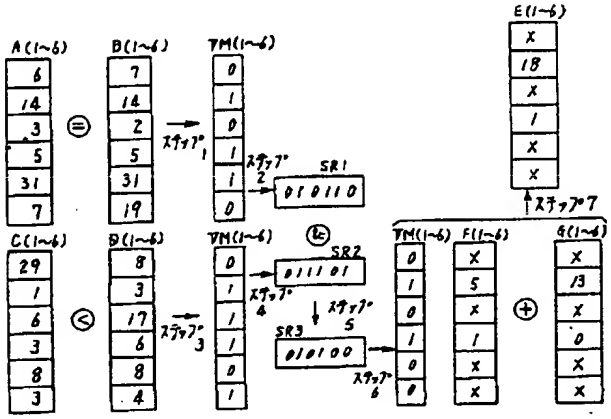
第 1 図

```
DO   10  I=1,N
   IF (
&        (A(I). EQ. B(I))        式(1)
&        .AND.
&        ((I). LT. D(I))         式(2)
&        )
&        E(I) = F(I)+G(I)        式(3)
10   CONTINUE
```

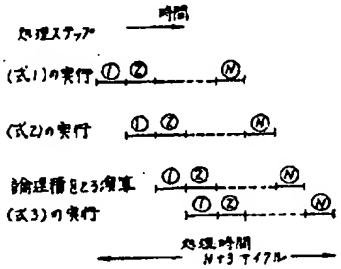
第 3 図



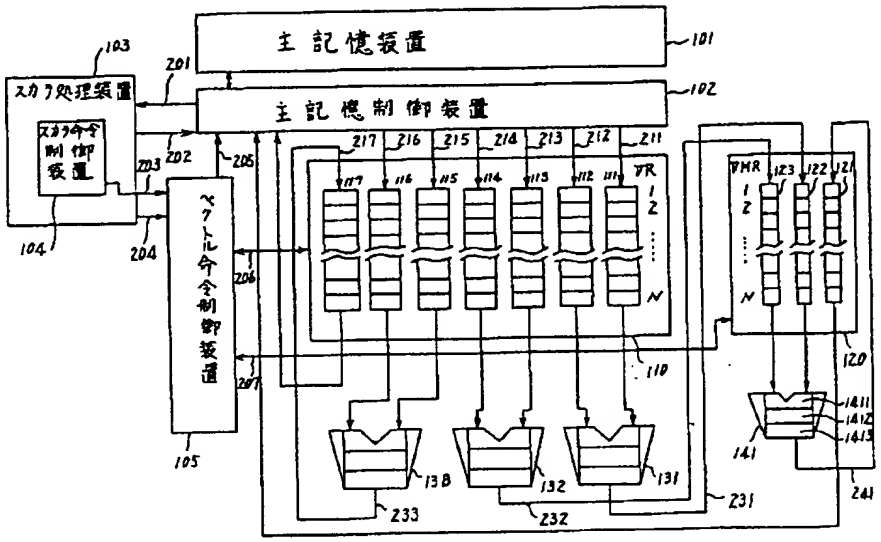
第 2 図



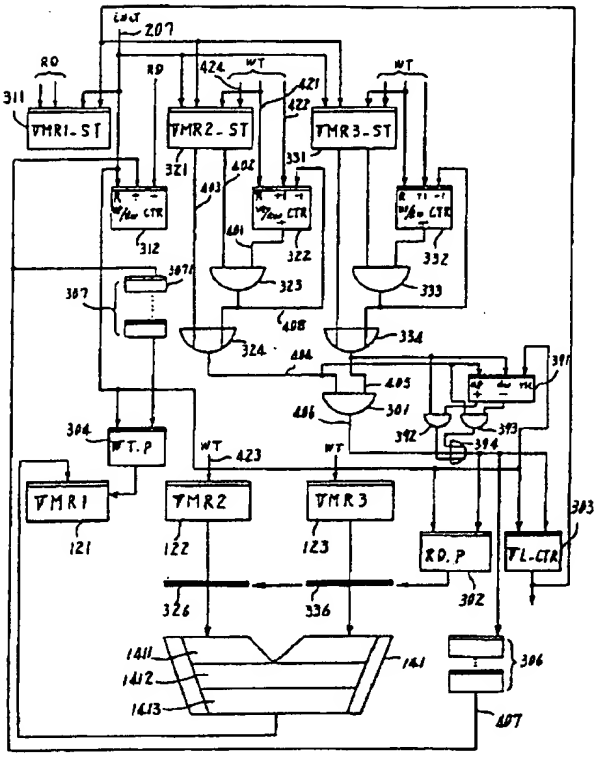
第 4 図



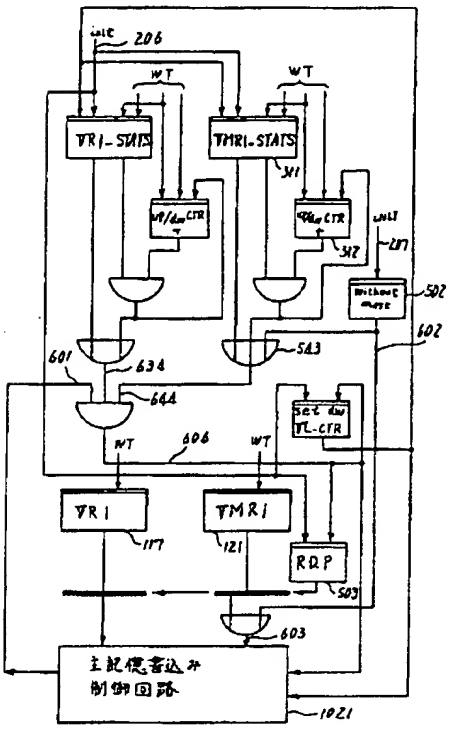
第 5 図



第 6 図


207
RD
WT
311
VMR1-ST
VMR2-ST
VMR3-ST
321
331
312
322
332
323
333
324
334
301
302
303
304
WT.P
VMR1
VMR2
VMR3
121
122
123
RD.P
VL-CTR
326
336
1411
1412
1413
161
306
407
423
424
421
422

第 7 図


206
WT
VR1-STATS
VMR1-STATS
341
342
207
502
503
601
602
634
644
606
VR1
VMR1
117
121
RD.P
603
主記憶書込み
制御回路
1021